A Commitment-based Communicative Act Library

Mario Verdicchio
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italia
mario.verdicchio@polimi.it

Marco Colombetti
Politecnico di Milano
Milano, Italia
Università della Svizzera Italiana
Lugano, Switzerland
marco.colombetti@polimi.it

ABSTRACT

The Agent Communication Language (ACL) proposed by the Foundation for Intelligent Physical Agents (FIPA) is the most complete attempt to create a universally accepted standard so far. Nevertheless, this standard shows some shortcomings which are probably hindering an even greater impact upon the scientific research dealing with multiagent systems. Although agreeing with the mainstream view that analyzes agent communication in terms of communicative acts, we part from FIPA's assumptions about the semantics, as we shift the focus from affecting communicating agents' mental states to modifying the commitments binding them to each other. We show that our commitment-based framework is powerful enough to allow for the main FIPA communicative acts and provides a semantics which overcomes some of the problems that are currently affecting the standard.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Languages, Theory

Keywords

Agent Communication Languages, Commitment, Speech Acts

1. INTRODUCTION

Agent Communication Languages (ACLs) play a very important role in the context of open multiagent systems, which must provide a standard communication framework that allows all participating agents to interact. The fact that we do not have an established standard yet has lead us to research for some unresolved issues that may have hindered the universal acceptance of the proposals put forward so far. Our analysis focuses on the proposal by the Foundation for Intelligent Physical Agents (FIPA) [2] because it has recently emerged as the best candidate to become an established standard. Some works in the literature, like [16], aimed at showing that several critical issues rise from expressing the language's semantics in terms of mental states and that instead turning to social states (i.e. commitments) would provide a way to solve some of these issues. We aim to show that this approach is a real alternative to mentalistic semantics, and we consider as a fundamental step to rewrite FIPA's Communicative Act Library [3] according to this perspective, that is, express FIPA communicative acts in terms of commitments between agents. Such a task has led us to classify them into four different categories, as follows:

1. the acts in this category (i.e. *inform* and *request*) are such that even if they are given a new semantics in terms of commitments, their illocutionary force (their point, i.e. informing and requesting, respectively) is not affected by such change;

2. the semantics of the acts that fall in this category (e.g. *propose*) is slightly changed when they are described in terms of commitments (for instance, a *propose* act is not defined as informing about one's intentions to perform a certain action, but as creating a proposal of a commitment to such action);

3. we put into this category those acts, like *request-when*, that do not express an illocutionary force, but a compound of illocutionary force and content; we suggest a way to redefine such acts (e.g. we define a *request-when* act as a *request* act with a temporal conditional content);

4. this last category is comprised of those acts, like *confirm* and *disconfirm*, that are not considered necessary in an approach which does not take mental states into account; enriching our model to include such acts is beyond the scope of this work.

This paper is organized as follows: Section 2 provides the formal apparatus our model is based upon; Section 3 illustrates a way to formalize commitments in a multiagent system; Section 4 defines FIPA communicative acts belonging to the first three categories above in terms of commitments; Section 5 finally draws our conclusions and illustrates the future directions of this work.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

2. THE FORMAL MODEL

As commitments deal with certain states of affairs that occur in time, we first need to provide some formal definitions about time, events, and actions. To do so, we provide new definitions which extend the temporal logic that is illustrated in [16] with a different notation to increase readability. Our starting point is CTL$^{\pm}$, a temporal language close to CTL*, which is a branching temporal logic including only future-directed temporal operators [1]. Past-directed operators do not increase the logic's expressiveness [8], but nevertheless they allow us to express some properties of computational systems in a far more succinct way [11]. In CTL$^{\pm}$, time is assumed to be discrete, with no start or end point, and branching only in the future. In the literature we can find temporal logic proposals that involve branching also in the past [14], but we prefer to rely on the idea of "historical necessity" [15], according to which agents have no possibility of changing the past, so that they are enabled to reason about alternatives or indeterminacy only with respect to the future.

2.1 The syntax

We call *sort set* a finite, nonempty set of elements, called *sorts*; a finite, possibly empty sequence of sorts is called a prototype. A CTL$^{\pm}$ language is a sextuple $\langle \Sigma, V, C, \Xi, \Pi, \theta \rangle$, where Σ is a sort set, V is a denumerable set of (individual) variables, C is an arbitrary set of (individual) constants, Ξ is an arbitrary set of functors, Π is an arbitrary set of predicates, and θ is a function that assigns a sort to every variable and every constant, and a prototype to every functor and every predicate. The set V of variables includes denumerable many variables for every sort.

For every sort σ, we define the set T_σ of *terms* of sort σ as follows:

- $x \in T_\sigma$ if $x \in V$ and $\theta(x) = \sigma$;
- $a \in T_\sigma$ if $a \in C$ and $\theta(a) = \sigma$;
- $f(t_1, ..., t_n) \in T_\sigma$ if $f \in \Xi$ and $\theta(f) = \langle \sigma, \theta(t_1), ..., \theta(t_n) \rangle$;
- nothing else belongs to T_σ.

The set A of *atomic formulae* is defined as follows:

- $(t_1 = t_2) \in A$ if $t_1, t_2 \in T_s$ for some $s \in \Sigma$;
- $P(t_1, ..., t_n) \in A$ if $P \in \Pi$, $\theta(P) = \langle \sigma_1, ..., \sigma_n \rangle$ and $t_i \in T_{\sigma_i}$ for $1 \le i \le n$;
- nothing else belongs to A.

The set Φ of CTL$^{\pm}$ *formulae* is such that:

- $A \subseteq \Phi$;
- $\neg\phi \in \Phi$ if $\phi \in \Phi$;
- $(\phi \wedge \psi) \in \Phi$ if $\phi, \psi \in \Phi$;
- $\forall x \phi \in \Phi$ if $x \in V$ and $\phi \in \Phi$;
- $\mathsf{Next}\phi$, $\mathsf{Pre}\phi \in \Phi$ if $\phi \in \Phi$;
- $(\phi \mathsf{Until} \psi)$, $(\phi \mathsf{Since} \psi) \in \Phi$ if $\phi, \psi \in \Phi$;
- $\mathsf{A}\phi$ if $\phi \in \Phi$;
- nothing else belongs to Φ.

The temporal operators Next (at the next state), Pre (at the previous state), Until, Since, and A (on all paths), are primitive. The formulae *true*, *false*, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, $(\phi \leftrightarrow \psi)$, and $\exists x \phi$ respectively abbreviate $\forall x(x = x)$, $\neg true$, $\neg(\neg\phi \wedge \neg\psi)$, $(\neg\phi \vee \psi)$, $((\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi))$, and $\neg\forall x \neg\phi$. As usual, $\phi[t/x]$ denotes the result of replacing all free occurrences of variable x in ϕ with term t. Formula $\mathsf{E}\phi$ abbreviates $\neg\mathsf{A}\neg\phi$. We also introduce these temporal operators, SomeFut (sometimes in the future), SomePast (sometimes in the past), AlwFut (always in the future), AlwPast (always in the past), Some (sometimes), and Alw (always) as abbreviations, as follows:

$$\mathsf{SomeFut}\phi \triangleq true\mathsf{Until}\phi; \quad \mathsf{SomePast}\phi \triangleq true\mathsf{Since}\phi;$$
$$\mathsf{AlwFut}\phi \triangleq \neg\mathsf{SomeFut}\neg\phi; \quad \mathsf{AlwPast}\phi \triangleq \neg\mathsf{SomePast}\neg\phi;$$
$$\mathsf{Some}\phi \triangleq \mathsf{SomeFut}\phi \wedge \mathsf{SomePast}\phi;$$
$$\mathsf{Alw}\phi \triangleq \mathsf{AlwFut}\phi \wedge \mathsf{AlwPast}\phi.$$

2.2 The semantics

A CTL$^{\pm}$ frame is a structure $F = \langle S, \pi \rangle$, where S is a set of states, and $\pi : S \rightarrow S$ is an injective function that associates to every state a unique predecessor. Function π is such that every state is the predecessor of at least one state. A *path* in F is an infinite sequence $p = \langle p_0, ..., p_n, ... \rangle$ of states, in which every element p_n of the sequence is the predecessor of p_{n+1} in F. The subsequence of p starting from element p_n is itself a path, which we denote with p^n; for every $n > 0$, we say that p^n is a *subpath* of p.

A *multidomain* $D = \{D_\sigma\}_{\sigma \in \Sigma}$ is a collection of mutually disjoint, nonempty domains of individuals. A *model* for CTL$^{\pm}$ is a triple $M = \langle F, D, i \rangle$, where F is a CTL$^{\pm}$ frame, D is a multidomain, and i is an interpretation function assigning:

- an individual $i(c) \in D_{\theta(c)}$ to every constant c;
- a function $i(s, f) : D_{\sigma_1} \times ... \times D_{\sigma_n} \rightarrow D_\sigma$ to every state s and every function f such that $\theta(f) = \langle \sigma, \sigma_1, ..., \sigma_n \rangle$;
- a relation $i(s, P) \subseteq D_{\sigma_1} \times ... \times D_{\sigma_n}$ to every state s and every predicate P such that $\theta(P) = \langle \sigma_1, ..., \sigma_n \rangle$.

An assignment of individuals to variables is a function $v : V \rightarrow D$ such that $v(x) \in D_{\theta(x)}$. Given assignment v, an assignment v' is an *x-variant* of v ($v \approx_x v'$, in symbols) if $v(y) = v'(y)$ for all $y \neq x$. The denotation of term t under an assignment v is defined as follows:

- $\delta_{M,v}(t) = v(t)$ if $t \in V$;
- $\delta_{M,v}(t) = i(t)$ if $t \in C$;
- $\delta_{M,v}(f(t_1, ..., t_n)) = i(s, f)(\delta_{M,v}(t_1), ..., \delta_{M,v}(t_n))$.

Denotations do not depend on paths, so that constants are rigid.

Let us define the conditions under which a formula is true in model M on path p under assignment v:

$M, p, v \models (t_1 = t_2)$ iff $\delta_{M,v}(t_1) = \delta_{M,v}(t_2)$;
$M, p, v \models P(t_1, ..., t_n)$ iff $\langle \delta_{M,v}(t_1), ..., \delta_{M,v}(t_n) \rangle \in i(p_0, P)$;
$M, p, v \models \neg\phi$ iff not $M, p, v \models \phi$;
$M, p, v \models (\phi \wedge \psi)$ iff $M, p, v \models \phi$ and $M, p, v \models \psi$;

$M,p,v \models \forall x\phi$ iff $M,p,v' \models \phi$ for all v' such that $v \approx_x v'$;
$M,p,v \models \mathsf{Next}\ \phi$ iff $M,p^1,v \models \phi$;
$M,p,v \models \mathsf{Pre}\ \phi$ iff for some path q, $q^1 = p$ and $M,q,v \models \phi$;
$M,p,v \models (\phi\ \mathsf{Until}\ \psi)$ iff for some n, $M,p^n,v \models \psi$ and for all m s.t. $0 \leq m \leq n$, $M,p^m,v \models \phi$;
$M,p,v \models (\phi\ \mathsf{Since}\ \psi)$ iff for some path q and for some n, $q^n = p$ and $M,q \models \psi$ and for all m s.t. $0 \leq m \leq n$, $M,q^m \models \phi$;
$M,p,v \models \mathsf{A}\phi$ iff for all q s.t. $q_0 = p_0$, $M,q,v \models \phi$.

Please note that the definitions of Until and Since deviate from the classical ones, in that they are inclusive of both states at the boundaries of the relevant interval. A formula is true in model M on path p if it is true in M on path p under all assignments:

$M,p \models \phi$ iff $M,p,v \models \phi$ for all v.

If ϕ is a closed formula (i.e. it contains no free occurrences of variables), then its truth value does not depend on variable assignments. Thus, ϕ is true in M on p under v if, and only if, it is true in M on p. Finally, a formula is valid if it is true on all paths of every model:

$\models \phi$

if and only if $M,p \models \phi$ for all models M and all paths p in the frame of M.

3. COMMITMENTS AND THEIR MANIPULATION

To define a logic of commitments we have to deal with agents, typed events that agents bring about, and sentences of a content language (CL) that are the content of their commitments. Thus, we need to introduce the relevant sorts in our language.

3.1 Events and actions

We assume that the set Σ of CTL$^{\pm}$ sorts contains at least the elements *event* (the sort of events), *agent* (the sort of agents), *eventtype* (the sort of event types), and *sentence* (the sort of CL sentences), and that the set P of predicates contains at least the elements *Happ*, *Actor*, *Type*, *Comm*, and *Prec*. To indicate that predicate *Happ*'s prototype is $\theta(Happ) = \langle event \rangle$, we write $Happ(event)$, and we use a similar notation for the other predicates:

- $Actor(event, agent)$;
- $Type(event, eventtype)$;
- $Comm(event, agent, agent, sentence)$;
- $Prec(event, agent, agent, sentence)$.

Intuitively, $Happ(e)$ means that event e has just happened, $Actor(e,x)$ means that event e, if it happened at all, has been brought about by agent x (we also say that x is the actor of e), and $Type(e,t)$ means that e is an event of type t. We assume that an event cannot happen more than once on the same path. This assumption is captured by the following event uniqueness axiom:

(EU) $Happ(e) \rightarrow \mathsf{PreAlwPast}\neg Happ(e) \wedge \mathsf{ANextAlwFut}\neg Happ(e)$.

Here we define the predicate $Done(event, agent, eventtype)$:

(DD) $Done(e,x,t) \triangleq Happ(e) \wedge Actor(e,x) \wedge Type(e,t)$.

The intuitive meaning of $Done(e,x,t)$ is that agent x has just brought about an event e of type t, or, that x has performed an action e of type t. We thus define actions as events that have an actor.

3.2 Commitments and precommitments

In our approach, we analytically define communicative acts (a special type of actions) in terms of changes at the level of social relations among agents. We take commitment to be a primitive concept that underlies the social structure of a multi-agent system, and describe communicative acts as actions brought about by an agent to affect the network of commitments that binds it to other agents. We thus introduce the *Comm* predicate: to state that a commitment holds at a state in which agent x (the debtor) is bound, relative to agent y (the creditor), to the fact that some proposition (the content) is true, we write

$M,p,v \models Comm(e,x,y,s)$.

The first argument of the *Comm* predicate, e, is the event that has brought about the state of affairs in which the commitment holds. The content of the commitment is a formula of a content language represented as a first-order term s of sort *sentence*, which is the fourth parameter of the *Comm* predicate. The semantics of CL sentences is provided by translating them into formulae of Φ. More formally, we define a function $\lfloor\ \rfloor : D_{sentence} \rightarrow \Phi$ such that, given a *sentence term* s, $\lfloor s \rfloor$ is the Φ formula it corresponds to. We also introduce a function $\lceil\ \rceil : \Phi \rightarrow D_{sentence}$ which, given a formula $\phi \in \Phi$, returns the relevant term $\lceil \phi \rceil \in D_{sentence}$.

Commitments that have been proposed but not yet accepted nor rejected are defined as *precommitments*. They are represented in the same way as commitments: the following formula holds when e has brought about a precommitment between two agents (the potential debtor x and the potential creditor y) to the truth of a sentence represented by s:

$M,p,v \models Prec(e,x,y,s)$.

In our approach, agent communication is brought about by means of message exchanges that under specific conditions count as commitment manipulation actions. We suppose that the set $D_{eventtype}$ contains the following event types, corresponding to five basic actions for manipulating commitments:

1. make-commitment: $mc(x,y,s)$;
2. make-precommitment: $mp(x,y,s)$;
3. cancel-commitment: $cc(e,x,y,s)$;
4. cancel-precommitment: $cp(e,x,y,s)$;
5. accept-precommitment: $ap(e,x,y,s)$.

The *mc* and *mp* event types have three parameters, x, y, and s, that correspond to the debtor, the creditor, and the content of the (pre)commitment that is being created.

The *cc*, *cp*, and *ap* event types have one more parameter e, that refers to the event that has brought about the (pre)commitment that is being cancelled or accepted. These are types of actions that agents can perform. For instance, to state that agent x has brought about an event e of making a commitment towards agent y with content s, we write as follows:

$$M, p, v \models Done(e, x, mc(x, y, s)).$$

We may use the 'm-dash' character to express existential quantification, as follows:

$Done(e, -, t) \triangleq \exists x Done(e, x, t)$;
$Done(-, -, t) \triangleq \exists e \exists x Done(e, x, t)$;
$Done(e, -, -)$ corresponds to the primitive $Happ(e)$.

Here are the axioms that describe the above mentioned types of commitment manipulation events in terms of their constitutive effects, that is, the state of affairs that are the case after a token of the given event type is performed.

These axioms feature the Z temporal operator, which represents the intuitive concept of "until and no longer" and is defined as follows:

$$\phi \mathsf{\ Z\ } \psi \triangleq \phi \mathsf{\ WeakUntil} \psi \wedge \mathsf{AlwFut}(\psi \rightarrow \mathsf{NextAlwFut} \neg \phi),$$
where
$$\phi \mathsf{\ WeakUntil\ } \psi \triangleq \mathsf{AlwFut\ } \phi \vee \phi \mathsf{\ Until\ } \psi.$$

$\phi \mathsf{\ Z\ } \psi$ is true if and only if in the future ψ never becomes true and ϕ is always true, or ϕ is true until ψ eventually becomes true and since then ϕ is no longer true.

(MC) $Done(e, -, mc(x, y, s)) \rightarrow$
$\mathsf{A}\ (Comm(e, x, y, s) \mathsf{\ Z\ } Done(-, -, cc(e, x, y, s)))$;
(MP) $Done(e, -, mp(x, y, s)) \rightarrow$
$\mathsf{A}\ (Prec(e, x, y, s) \mathsf{\ Z\ } (Done(-, -, ap(e, x, y, s)) \vee Done(-, -, cp(e, x, y, s))))$;
(AP) $Done(e', -, ap(e, x, y, s)) \rightarrow$
$\mathsf{A}(Comm(e', x, y, s) \mathsf{\ Z\ } Done(-, -, cc(e', x, y, s)))$.

Axiom MC (Make Commitment) states that if an agent (not necessarily x or y) performs an action of making a commitment with x as the debtor, y as the creditor, and s as the content, then on all paths x is committed, relative to y, to the truth of s, until an agent possibly cancels such a commitment, after which the commitment no longer exists. Axiom MP (Make Precommitment) is analogous to MC, and it deals with the creation of a precommitment. Axiom AP (Accept Precommitment) entails that if an agent performs the action of accepting a precommitment brought about by event e with x, y, and s respectively as debtor, creditor, and content, then such acceptance brings about on all paths a commitment of x, relative to y, to the truth of s, which will hold until it is possibly cancelled. There are no specific axioms for the actions of cancelling a precommitment (*cp*) or a commitment (*cc*), because the analytical effects of these commitment manipulations are already illustrated in the axioms dealing with other actions.

Commitments are said to be *fulfilled* and *violated* when their content is *settled true* and *false*, respectively. Before dealing with the truth conditions of formulae, we must take the following considerations into account. Firstly, the truth of a sentence including temporal qualifications at a given state (namely, the *point of reference* [10]) can be evaluated only if we know the state at which the sentence has been uttered (the *point of speech*). Moreover, branching time brings in a phenomenon known as *contingent future*, which means that at a given point of reference it may be still undetermined if a sentence is going to be true or false (e.g., "it will rain until 6:00", stated while it is raining at 4:00). In such a case, a formula is said to be *unsettled*, and the relevant commitment is *pending*. The truth conditions of CL sentences are thus formalized as follows:

(DT) $True(e, s) \triangleq \mathsf{ASomePast}(Happ(e) \wedge \lfloor s \rfloor)$,
(DF) $False(e, s) \triangleq \mathsf{ASomePast}(Happ(e) \wedge \neg \lfloor s \rfloor)$,
(DU) $Unset(e, s) \triangleq \mathsf{ASomePast} Happ(e) \wedge \neg True(e, s) \wedge \neg False(e, s)$.

The truth conditions of sentence s are given with respect to an event e, which does not necessarily correspond to the event of uttering s. As event e is used to set a well-defined temporal reference by which we can evaluate the truth of s, all these definitions rely on the event uniqueness axiom. We then have the following definitions:

(DL) $Fulf(e, x, y, s) \triangleq Comm(e, x, y, s) \wedge True(e, s)$,
(DV) $Viol(e, x, y, s) \triangleq Comm(e, x, y, s) \wedge False(e, s)$,
(DP) $Pend(e, x, y, s) \triangleq Comm(e, x, y, s) \wedge Unset(e, s)$.

3.3 Action expressions

So far, we have dealt with commitments with a generic content term s, but we may want to focus on more specific contents, dealing with future actions performed by agents. To do so, let us first introduce some derived temporal operators, which will enable us to write more synthetic formulae:

$$\phi \mathsf{\ Before\ } \psi \triangleq \neg(\neg \phi \mathsf{\ WeakUntil\ } \psi);$$
$$\phi \mathsf{\ AsSoonAs\ } \psi \triangleq (\psi \rightarrow \phi) \mathsf{\ WeakUntil\ } \psi.$$

Formula $\phi \mathsf{\ AsSoonAs\ } \psi$ holds when at the first state at which ψ is true, also ϕ is the case. In other words, ϕ is true as soon as ψ is (possibly) true. We define a subdomain $D_{action} \subseteq D_{sentence}$ of *action expressions*, that is, terms corresponding to a particular subset of CL sentences. We have identified two action expression schemata, which are expressive enough to allow for a vast range of formulae dealing with action performances. Let d, s_λ, s_ω, and s_χ be terms of $D_{sentence}$. In particular, let d correspond to a formula representing the performance of an action of a certain type by an agent: $\lfloor d \rfloor = Done(-, t_1, t_2)$, where $t_1 \in T_{agent}$ and $t_2 \in T_{eventtype}$. We then define the first action expression schema $\alpha_\forall$ as follows:

$$\alpha_\forall = [s_\lambda, s_\omega | s_\chi] d$$
$$\lfloor \alpha_\forall \rfloor = ((\lfloor s_\chi \rfloor \rightarrow \lfloor d \rfloor) \mathsf{WeakUntil} \lfloor s_\omega \rfloor) \mathsf{AsSoonAs} \lfloor s_\lambda \rfloor.$$

We have $M, p, v \models \lfloor \alpha_\forall \rfloor$ if and only if in the sequence of states on path p which begins at the first occurrence of $\lfloor s_\lambda \rfloor$ and ends at the subsequent state at which $\lfloor s_\omega \rfloor$ is the case, every time $\lfloor s_\chi \rfloor$ holds, then $\lfloor d \rfloor$ is true. The notation with square brackets has already been adopted in [18], and it was originally inspired by [9]. We define the other action expression schema, $\alpha_\exists$, as follows:

$$\alpha_\exists = \langle s_\lambda, s_\omega | s_\chi \rangle d$$
$$\lfloor \alpha_\exists \rfloor = ((\lfloor s_\chi \rfloor \mathsf{Before} \lfloor s_\omega \rfloor) \rightarrow ((\lfloor s_\chi \rfloor \wedge \lfloor d \rfloor) \mathsf{Before} \lfloor s_\omega \rfloor)) \mathsf{\ AsSoonAs} \lfloor s_\lambda \rfloor.$$

If $\alpha_\forall$ is loosely based on the idea of universal quantification, $\alpha_\exists$ deals with existential quantification, in that $\lfloor\alpha_\exists\rfloor$ is true on a path where, in the interval identified by the subsequent occurrences of $\lfloor s_\lambda\rfloor$ and $\lfloor s_\omega\rfloor$, $\lfloor d\rfloor$ is true at a state at which $\lfloor s_\chi\rfloor$ holds, if it is ever the case. We will refer to a generic action expression, whether universal or existential, by α. Given an action expression α, we denote with $agent(\alpha)$ the agent that is designated as the actor of the action to be performed.

4. COMMUNICATIVE ACTS

An institutional action is defined within the context of an artificial institution, a set of shared rules that regulate the management of a fragment of social reality [13, 7], including multiagent systems. It is performed through the execution of some lower level act that *conventionally* counts as a performance of the institutional action; an example is provided by communicative acts, which are performed by executing lower level acts of message exchange. Institutional actions thus require a set of conventions for their execution. We adopt the view according to which the commitment manipulation actions described in the previous section are institutional actions that are conventionally realized by the exchange of messages.

4.1 Basic communicative acts

In our approach, we call *basic* those communicative acts that map directly onto a commitment manipulation action. For every kind of message we introduce a functor that specifies the type of the action that an agent performs when exchanging such a message. This approach is illustrated by the following example. Suppose that a message is sent to agent y to inform y that $\lfloor s\rfloor$ is the case. The exchange of such a message is an event of type $inform(y, s)$, where *inform* is a two-place functor denoting the type of the message, y denotes the receiver of the message, and s is its content. When event e is an exchange of a message of type *inform* and content s, sent by agent x to agent y, the following formula holds:

$$Done(e, x, inform(y, s)).$$

This event, under given conditions which we illustrate later on, implies the performance of a commitment manipulation action. The semantics of the message is defined as the effect that exchanging such a message has on the network of commitments binding the sender and the receiver. The correspondence between the message exchange event type and the commitment manipulation action type relies on a relation that is formally described by the formula below,

$$Conv(x, t, t'),$$

which means that an action of type t performed by agent x corresponds to an action of type t' in accordance to a convention established in the communication framework. Here we define a basic set of communicative acts by means of which agents carry out the commitment manipulation actions. Each communicative act is accompanied by a possibly empty set of conditions that must hold to make the message exchange count as a commitment manipulation. These conditions deal with the agent designated to perform an action (e.g., if x makes a request to y, y must be the performer of the requested action), with the creators of precommitments (e.g., if x accepts a proposal by exchanging a message with y, y must be the issuer of such proposal), and with some presuppositions about the existence of precommitments (i.e., x cannot reject a proposal that has not been made). This is formally stated by the following axiom:

$$\text{(CO)}\ Done(e, x, t) \wedge Conv(x, t, t') \wedge \Psi \rightarrow Done(e, x, t'),$$

where Ψ is to be understood as the conjunction of the formulae indicated in the fourth column of the table in Figure 1 for each communicative act type. A comparison between our approach and another work dealing with the notion of convention [6] can be found in [17].

message type	Conv(x,t,t’)		additional conditions
	t	*t’*	
inform	*inform(y,s)*	*mc(x,y,s)*	
request	*request(y, a)*	*mp(y,x,a)*	*y=agent(a)*
agree	*agree(y,(e,x,y,a))*	*ap(e,x,y,a)*	*x=agent(a)* *Actor(e,y)* Pre *Prec(e,x,y,a)*
propose	*propose(y,a)*	*mp(x,y,a)*	*x=agent(a)*
accept-proposal	*accept-proposal(y,(e,y,x,a))*	*ap(e,y,x,a)*	*y=agent(a)* *Actor(e,y)* Pre *Prec(e,y,x,a)*
refuse	*refuse(y,(e,x,y,a))*	*cp(e,x,y,a)*	*x=agent(a)* *Actor(e,y)* Pre *Prec(e,x,y,a)*
reject-proposal	*reject-proposal(y,(e,y,x,a))*	*cp(e,y,x,a)*	*y=agent(a)* *Actor(e,y)* Pre *Prec(e,y,x,a)*
cancel	*cancel(y,(e,y,x,a))*	*cc(e,y,x,a)*	*y=agent(a)* Pre *Comm(e,y,x,a)*

Figure 1: FIPA communicative acts as conventions to perform commitment manipulations.

4.2 Derived communicative acts

Derived communicative acts are defined in terms of the above mentioned basic communicative acts. We can distinguish two kinds of derivation. In the case of the *request-when* and *request-whenever* acts, we deal with derivation by *content specialization*, in that we define them as a *request* act with a specialized content. We thus part from FIPA's specifications, which define these acts in terms of an inform act. “Request-when allows an agent to inform another agent that a certain action should be performed as soon as a given precondition... becomes true [3].” A very similar description is provided for the request-whenever act. We avoid defining a request-like act in terms of a basic act of informing since we keep in mind a fundamental concept from Speech Act Theory (i.e. the *direction of fit* [12]), according to which a request has a *world-to-word* direction (we want the world to be like what we ask) while an inform act has a *word-to-world* direction (what we state must reflect the current state of affairs). Besides, we think that these definitions clash with the fact that the simple request is considered as a primitive act in FIPA's specification. If specialized requests can be defined in terms of an inform act, we should also be able to define the simple request in such terms. In our model, the *request-when* and *request-whenever* acts are formally defined as follows:

$$request\text{-}when(y, t_1, s_1) \triangleq request(y, \alpha_1),$$
$$\alpha_1 = \langle \lceil true \rceil, s_1 | s_1 \rangle \lceil Done(-, y, t_1) \rceil;$$
$$request\text{-}whenever(y, t_2, s_2) \triangleq request(y, \alpha_2),$$
$$\alpha_2 = [\lceil true \rceil, \lceil false \rceil | s_2] \lceil Done(-, y, t_2) \rceil.$$

As it can be easily shown that

$$\lfloor \langle \lceil true \rceil, s_1 | s_1 \rangle \lceil Done(-, y, t_1) \rceil \rfloor \leftrightarrow Done(-, y, t_1) \text{ AsSoonAs } \lfloor s_1 \rfloor,$$

by performing a message exchange of type $request\text{-}when(y, t_1, s_1)$, an agent requests y to perform an act of type t_1 as soon as $\lfloor s_1 \rfloor$ holds. Similarly, as we have

$$\lfloor [\lceil true \rceil, \lceil false \rceil | s_2] \lceil Done(-, y, t_2) \rceil \rfloor \leftrightarrow \text{AlwFut}(\lfloor s_2 \rfloor \rightarrow Done(-, y, t_2)),$$

a $request\text{-}whenever(y, t_2, s_2)$ act consists of a request to y to perform a t_2 action every time $\lfloor s_2 \rfloor$ is the case.

Following the FIPA specifications, we define the *inform-if* act as a 'macro' act to inform whether a sentence is true or not. In this case, we deal with derivation by *macro composition*, as we define an *inform-if* act as a disjunction of mutually exclusive *inform* acts, as follows.

$$Done(e, x, inform\text{-}if(y, s)) \triangleq$$
$$Done(e, x, inform(y, s)) \underline{\vee} Done(e, x, inform(y, \lceil \neg \lfloor s \rfloor \rceil)),$$

where $\phi \underline{\vee} \psi \triangleq (\phi \vee \psi) \wedge \neg(\phi \wedge \psi)$. We then define a *query-if* act by content specification as a *request* for an *inform-if* act, as follows:

$$Done(e, x, query\text{-}if(y, s)) \triangleq Done(e, x, request(y, \alpha)),$$
$$\alpha = \langle \lceil true \rceil, \lceil false \rceil | \lceil true \rceil \rangle \lceil Done(-, y, inform\text{-}if(x, s)) \rceil,$$

where the temporal qualification $\langle \lceil true \rceil, \lceil false \rceil | \lceil true \rceil \rangle \phi$, which is equivalent to SomeFutϕ, may be specified in order to introduce deadlines for the *inform-if* act, as we have $\langle \lceil true \rceil, \lceil \psi \rceil | \lceil true \rceil \rangle \phi \leftrightarrow \phi$ Before ψ.

Failure and *not-understood* are two more acts that can be defined as *inform* acts with a specific content, expressed in terms of predicates dealing with *attempts* and *message decoding* whose definition lies beyond the scope of this work.

4.3 Communicative acts with referential operators

Following FIPA's specifications [5], we introduce three *referential operators*, *any*, *iota*, and *all*, to create referential terms like $(any\ x\ f)$, $(iota\ x\ f)$, and $(all\ x\ f)$ (with $f \in D_{sentence}$) which are to be read as "any x", "the x", and "all the x" such that $\lfloor f \rfloor$ is true . We will not provide a formal definition of such terms, in that in our approach they are used only as a notation to distinguish one referential act from another. We assume that there exists a sort *URI* of *uniform resource identifiers*, which identify every object in multidomain D with a unique name, and a function $uri : D \rightarrow D_{URI}$ that returns the URI of every element in D. URIs are assumed to be self-referential. Given a referential term r, we define the *inform-ref* act as a specialization of an *inform* act, as follows:

$$inform\text{-}ref(y, r) \triangleq inform(y, s),$$

where s corresponds to a specific formula in accordance with r, as follows:

if $r = (any\ x\ f)$, then
$\lfloor s \rfloor = \lfloor f \rfloor [k/x] \ \wedge \ uri(k) = n;$
if $r = (the\ x\ f)$, then
$\lfloor s \rfloor = \lfloor f \rfloor [k/x] \wedge uri(k) = n \wedge \forall z(\lfloor f \rfloor [z/x] \rightarrow z = k);$
if $r = (all\ x\ f)$, then
$\lfloor s \rfloor = \bigwedge_i (\lfloor f \rfloor [k_i/x] \wedge (uri(k_i) = n_i)) \wedge \forall z(\lfloor f \rfloor [z/x] \rightarrow \bigvee_i (z = k_i)).$

We then define the *query-ref* act as a *request* for an *inform-ref* act, as below:

$$Done(e, x, query\text{-}ref(y, r)) \triangleq Done(e, x, request(y, \alpha)),$$
$$\alpha = \lceil \text{SomeFut} Done(-, y, inform\text{-}ref(x, r)) \rceil.$$

As stated before, we have that:

$$\text{SomeFut}\phi \leftrightarrow \langle \lceil true \rceil, \lceil false \rceil | \lceil true \rceil \rangle \phi.$$

4.4 The Call For Proposal act

Let us have a closer look at the logical model (and at the advantages of an approach based on commitments rather than on mental states) while illustrating the *cfp* (call for proposal) act. Like FIPA, we define a *cfp* act as a *query-ref* act with a specific content, as follows:

$$Done(e, x, cfp(y, \tau)) \triangleq Done(e, x, query\text{-}ref(y, any\ w\ \alpha)),$$
$$\alpha = \langle \lceil Done(-, x, pay(y, w)) \rceil, \lceil Deadline \rceil | \lceil true \rceil \rangle \lceil Done(-, y, \tau) \rceil.$$

Considering also the *query-ref* act definition, we can see that a *cfp* act boils down to x asking y what is the sum w that x has to pay to y to have service τ done by y before a certain deadline (the $Done(-, x, pay(y, w))$ formula can be easily generalized or adapted to different application domains). Let us analyze how the commitments between two agents exchanging such a message evolve on a path p of a model M under an assignment v:

1. $M, p, v \models Done(e, x, cfp(y, \tau))$ (hypothesis)
2. $M, p, v \models Done(e, x, query\text{-}ref(y, any\ w\ \alpha))$ (1, *cfp* def)
3. $M, p, v \models Done(e, x, request(y, \alpha'))$ (2, *query-ref* def)
3°. $\lfloor \alpha' \rfloor = \text{SomeFut} Done(-, y, inform\text{-}ref(x, any\ w\ \alpha))$
4. $M, p, v \models Done(e, x, mp(e, y, x, \alpha'))$ (3,CO)
5. $M, p, v \models Prec(e, y, x, \alpha')$ (4, MP)

A *cfp* act to y by x thus leads to the creation of a precommitment of y towards x to perform an *inform-ref* act (we could also specify a deadline for such performance). Let us first suppose that y refuses such a precommitment (on a path p' that is a subpath of p, $\exists n(p' = p^n)$):

6′. $M, p', v \models Done(e', y, refuse(x, (e, y, x, \alpha')))$ (hyp)
7′. $M, p', v \models Done(e', y, cp(e, y, x, \alpha'))$ (6′, CO)
8′. $M, p', v \models \text{AAlwFut}\neg Prec(e, y, x, \alpha')$ (7′, MP)

As a result, the call for proposal has been turned down, and the relevant precommitment does not exist anymore. Let us show an alternative course of events on another subpath p'':

6″. $M, p'', v \models Done(e'', y, agree(x, (e, y, x, \alpha')))$ (hyp)
7″. $M, p'', v \models Done(e'', y, ap(e, y, x, \alpha'))$ (6″, CO)
8″. $M, p'', v \models \neg Prec(e, y, x, \alpha') \wedge Comm(e'', y, x, \alpha')$ (7″, AP)

An *agree* act by y turns the precommitment into a commitment. Let us suppose that later on, on a subpath p''' $(\exists m(p''' = p''^m))$, y informs x about the sum k that y requires for service τ (we omit the *uri* function in the content of the *inform* message):

$9''$. $M, p''', v \models Done(e''', y, inform(x, \lceil \lfloor \alpha \rfloor [k/w] \rceil))$ (hyp)
$10''$. $M, p''', v \models Done(e''', y, inform\text{-}ref(x, any\ w\ \alpha))$ ($9''$, *inform-ref* def)
$11''$. $M, p''', v \models \mathsf{ASomePast}(Happ(e'') \wedge \lfloor \alpha' \rfloor)$ ($3°$, $6''$, $10''$)
$12''$. $M, p''', v \models True(e'', \alpha')$ ($11''$, DT)
$13''$. $M, p''', v \models Fulf(e'', y, x, \alpha')$ ($8''$, $12''$, DL)

By performing an *inform-ref*, y fulfills a commitment, but such act, as it consists of an *inform* message exchange, also creates another commitment, as follows:

$14''$. $M, p''', v \models Done(e''', y, mc(y, x, \lceil \lfloor \alpha \rfloor [k/w] \rceil))$ ($9''$, CO)
$15''$. $M, p''', v \models Comm(e''', y, x, \lceil \lfloor \alpha \rfloor [k/w] \rceil)$ ($14''$, MC)

Agent y is thus committed to provide service τ before a specific deadline as soon as x pays k.

5. CONCLUSIONS AND FUTURE WORK

The *cfp* example illustrates the advantages of our approach with respect to FIPA's. The FIPA standard does not provide any mechanism to verify the fulfillment of the agents' commitments, if not relying on *inform* messages by the agents themselves, as in the FIPA Contract Net protocol [4]. Such *inform* acts do not entail the completion of a requested task, but provide only a snapshot of some of the *beliefs* of the messages' sender. Such beliefs reflect an actual state of affairs in which the task has been carried out only under specific assumptions about the agents' internal architecture, which we cannot afford if we aim at creating *open* multiagent systems. On the contrary, in our model every message brings about changes in the social reality that underlies the multiagent system: precommitments are created, cancelled, turned into commitments throughout the message exchange process. As (pre)commitments are public and reflect an objective state of affairs between agents, our model naturally provides a method to verify whether every agent has fulfilled its own duties.

In this work, we have treated referential operators simply as a matter of notation, making *inform-ref*, *query-ref*, and *cfp* acts rely on fairly complex content language sentences. We think that this solution may be changed in the future, when we tackle the *proxy* and *propagate* acts, whose definition seems to require some further investigation about the topic of *reference*.

6. REFERENCES

[1] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. MIT Press, Cambridge, MA, 1990.

[2] FIPA. Agent Communication Language Specifications. Specification, Foundation for Intelligent Physical Agents, http://www.fipa.org/repository/aclspecs.html, 2002.

[3] FIPA. Communicative Act Library Specification. Specification, Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00037/, 2002.

[4] FIPA. FIPA Contract Net Interaction Protocol Specification. Specification, Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00029/, 2002.

[5] FIPA. FIPA SL Content Language Specification. Specification, Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00008/, 2002.

[6] A. Jones and X. Parent. Conventional signalling acts and conversation. In F. Dignum, editor, *Advances in Agent Communication, International Workshop on Agent Communication Languages, ACL 2003, Melbourne, Australia, July 14, 2003*, volume 2922 of *Lecture Notes in Computer Science*. Springer, 2004.

[7] A. Jones and M. J. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):429–445, 1996.

[8] F. Laroussinie and P. Schnoebelen. A hierarchy of temporal logics with past. *Theoretical Computer Science*, 148(2):303–324, 1995.

[9] A. U. Mallya, P. Yolum, and M. P. Singh. Resolving commitments among autonomous agents. In F. Dignum, editor, *Advances in Agent Communication, Proceedings of the Interbational Workshop on Agent Communication Languages (ACL 2003)*, volume 2922 of *Lecture Notes in Artificial Intelligence*, pages 166–182. Springer, 2004.

[10] H. Reichenbach. *Elements of Symbolic Logic*. MacMillan, New York, NY, 1947.

[11] M. Reynolds. More past glories. In *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science (LICS'00)*, pages 229–240. IEEE Comp. Soc. Press, 2000.

[12] J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, UK, 1969.

[13] J. R. Searle. *The construction of social reality*. Free Press, New York, 1995.

[14] C. Stirling. Modal and temporal logics. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Philosophical Logic, Volume 2*, pages 477–563. Oxford University Press, Oxford, England, 1992.

[15] R. Thomason. Combinations of tense and modality. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Vol II: Extensions of Classical Logic*, pages 135–165. Reidel, Dordrecht, The Netherlands, 1984.

[16] M. Verdicchio and M. Colombetti. A logical model of social commitment for agent communication. In J. S. Rosenschein, T. Sandholm, M. J. Wooldridge, and M. Yokoo, editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 03)*, pages 528–535. ACM Press, 2003.

[17] M. Verdicchio and M. Colombetti. A logical model of social commitment for agent communication. In F. Dignum, editor, *Advances in Agent Communication, International Workshop on Agent Communication Languages, ACL 2003, Melbourne, Australia, July 14, 2003*, volume 2922 of *Lecture Notes in Computer Science*. Springer, 2004.

[18] M. Verdicchio and M. Colombetti. Dealing with time in content language expressions. In R. M. van Eijk, M.-P. Huget, and F. Dignum, editors, *Agent Communication, International Workshop on Agent Communication, AC 2004, New York, NY, USA, July 19, 2004, Revised Selected and Invited Papers*, volume 3396 of *Lecture Notes in Computer Science*. Springer, 2005.